Thermal-Inertial Odometry for Autonomous Flight Throughout the Night

Jeff Delaune, Robert Hewitt, Laura Lytle, Cristina Sorice, Rohan Thakker and Larry Matthies

Abstract— **Thermal cameras can enable autonomous flight at night without GPS. However, image-based navigation in the thermal infrared spectrum has been researched significantly less than in the visible spectrum. In this paper, we demonstrate closed-loop controlled outdoor flights at night on a quadrotor. Our state estimator can tightly couple inertial data with either thermal images at nighttime, or visual images at daytime. It is integrated in an autonomy framework for motion planning and control, which runs in real time on a standard embedded computer. We analyze thermal-inertial odometry performance extensively from sunset to sunrise, for various thermal non-uniformity levels, and compare it to visual-inertial odometry at daytime.**

I. INTRODUCTION

The next generation of autonomous drones requires more than GPS navigation. There is no GPS reception underground or on other planets. GPS signal may be degraded in urban canyons, unavailable indoors, jammed in military applications, or simply not accurate enough for some uses.

Visual-Inertial Odometry (VIO) has become a popular navigation alternative, as it leverages the accuracy of image geometric constraints while gaining metric scale, robustness and high estimation rate from an Inertial Measurement Unit (IMU). VIO can run in real time on lightweight drones thanks to the computational cost efficiency of modern Simultaneous Localization And Mapping (SLAM) algorithms [1], and the increased performance of embedded computers.

Thermal cameras capture the infrared radiation emitted by all objects with a temperature above the absolute zero in their field of view [2]. With the recent progress in increased resolution, and decreased size, weight, power and cost of these imagers, Thermal-Inertial Odometry (TIO) now appears as a promising approach to fly at night without GPS nor the cost of carrying a lidar sensor, or the range constraint of illumination-aided solutions. Unlike VIO, which operates with visible spectrum cameras, TIO has not yet been the subject of much literature.

The contrast of thermal infrared imagery varies throughout the day and night as objects with different thermal inertia heat and cool at different rates. The lowest contrast tends to occur right around dusk during thermal crossover [3], and at dawn when objects have completely cooled down. Therefore it is important to characterize TIO performance throughout the night. The uniformity of an uncooled thermal camera image over a uniform-temperature scene tends to decrease over time. Although modern cameras have factory-calibrated non-uniformity corrections, manufacturers still recommend to periodically perform a Flat-Field Correction (FFC) for optimal quality [4]. Since FFCs wash out the entire image for ~ 0.25 s, it is important to evaluate how TIO performance varies as a function of the time since the last FFC and when they are required.

This paper aims to demonstrate and characterize the performance of autonomous navigation for flight at night with thermal imagers. Our contributions are:

- a state estimator able to perform either TIO or VIO in flight,
- outdoor flight demonstration with TIO and closed-loop controls at night, as shown in Figure 1,
- an evaluation of TIO performance with respect to time of night,
- an evaluation of TIO performance with respect to time since FFC.

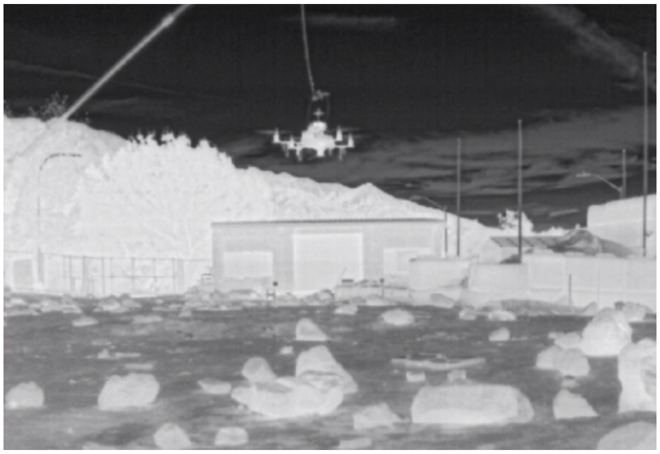


Fig. 1: Thermal view of a fully-autonomous quadrotor flight at night using the proposed thermal-inertial odometry, state machine and controls framework.

Section II discusses the relevant literature in thermal and visual-inertial odometry. Section III describes the common thermal and visual front end, along with the state estimation architecture. Section IV discusses the hardware setup, software parameters, control framework as well as the performance of our system in closed-loop controlled flights.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

The authors are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. {jeff.h.delaune, robert.a.hewitt, cristina.e.sorice, rohan.a.thakker, lhm}@jpl.nasa.gov, lalytle@usc.edu

Section V compares the TIO performance versus time of day, time since FFC, and VIO.

II. Related Work

A. Visual-inertial odometry

Soon after real-time visual SLAM became possible on consumer laptops with PTAM [5], the algorithms were adapted to work with flight trajectories and loosely-coupled with inertial sensors onboard drones [6]. Even though progress continues to be made with visual SLAM, VIO has seen improved results with tightly-coupled approaches [7]. Although tightly-coupled VIO is usually more computationally expensive than loosely-coupled, it is also usually more accurate and can rely on IMU integration more efficiently in degraded image conditions.

Approaches can be distinguished as filtering [8], [9] versus non-linear optimization [10], [11]. In either case, indirect approaches optimize on feature coordinates [8], [10] while direct ones optimize on image intensity values [9], [11]. None of these state-of-the-art VIO approaches have been characterized on thermal night imagery.

B. Thermal odometry

Thermal-based odometry is a much smaller body of literature than its visible spectrum spectrum counterpart. Due to the obvious perception challenge, it is interesting to identify how these approaches process thermal information.

[12] tracks Harris corners through Normalized Cross-Correlation (NCC) to constrain the proprioceptive odometry of a car at night [13]. [14] uses optical flow combined with road plane segmentation and FFC management. [15] uses SIFT keypoints and descriptors in both visible and thermal images [16]. [17] tracks ORB features in both RGB and thermal images [18]. [19] tracks GFTT and FAST features with KLT [20], [21]. It uses SURF descriptors to match data before and after a FFC event [22]. [23] also tracks SURF features but with FREAK descriptors in stereo images for a UAV [24]. [25] uses direct measurements in 14-bit monocular images on a UAV.

All these papers demonstrate performance on par with daytime VIO, using standard computer vision techniques. This would suggest that thermal image processing is not more challenging than vision. However, extensive performance analysis is still missing from the literature. Only [15] performs analysis over various types of environments. To the best of our knowledge, no existing literature has studied performance for various times of the night, or various times since FFC.

III. Thermal / Visual Inertial Odometry

Our odometry scheme applies image feature constraints to the state propagated with inertial measurements. The architecture is illustrated in Figure 2 and is based on an Extended Kalman Filter (EKF). It is is similar to state-of-the-art tightly-coupled visual-inertial approaches reviewed in Subsection II-A.

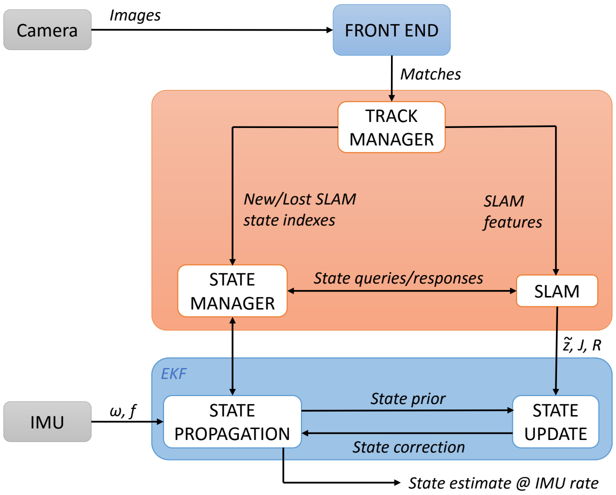


Fig. 2: Tightly-coupled estimator architecture capable of either thermal or visual-inertial odometry. The EKF state is propagated using the angular rates $\boldsymbol{\omega}$ and specific force $\boldsymbol{f}$ from the IMU. The SLAM module provides the residual $\tilde{\boldsymbol{z}}$, Jacobian $\boldsymbol{J}$ and covariance matrix $\boldsymbol{R}$ of the feature image coordinate measurements for the EKF update.

The front end matches features frame to frame from either a thermal or a visual camera. The best features are inserted in the EKF state vector to perform thermal or visual-inertial SLAM, respectively.

A. Front end

We detect FAST features in the image [20], and track them with the pyramidal implementation of Kanade-Lucas-Tomasi (KLT) algorithm by [26], [21]. We initially investigated feature tracking with NCC [27], and adaptive KLT [28] using the implementation in [29]. We found that the standard KLT runs faster than the other methods while maintaining similar tracking accuracy.

We estimate the camera rotation from image 1 to image 2 by integrating the angular velocity measurements coming from the gyroscopes and use it to initialize KLT for each feature according to

$$\begin{bmatrix}\mathbf{d}_2\\0\end{bmatrix} = \mathbf{q}_{21}\begin{bmatrix}\mathbf{d}_1\\0\end{bmatrix}\mathbf{q}_{21}^{-1}, \tag{1}$$

where $\mathbf{q}_{21}$ is the rotation quaternion, $\mathbf{d}_1$ and $\mathbf{d}_2$ are the direction vectors in camera frame 1 and 2, respectively.

At feature detection, a neighborhood parameter is defined in pixels to ensure feature are not too close to each other. When the number of feature matches falls below a threshold, FAST features are re-detected on the previous image and matched with the current one. Existing matches are preserved in this process. After matching features, Random Sampling and Consensus (RANSAC) [30] with a fundamental matrix model is used to remove outliers.

We call *tracks* the list of all image observations of a specific feature through time. Although all matches are organized into tracks, only a limited number of them are used for

SLAM to ensure the computational cost remains low. A track manager is responsible for selection of these tracks. It divides the image in tiles and ensures the number of SLAM features is similar between tiles. This spreads out measurements across the field-of-view (FOV) to improve pose estimation. In each tile, at first the features with the highest FAST score are selected. When a SLAM slot becomes available in a tile, we select the longest track from the pool of available ones, and the one with the highest FAST score when several tracks have the same length.

B. State estimation

Our state vector $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x_I}^T & \boldsymbol{x_V}^T\end{bmatrix}^T$ can be divided between the states related to the IMU $\boldsymbol{x_I}$, and those related to the camera $\boldsymbol{x_V}$.

The inertial states

$$\boldsymbol{x}_I = \begin{bmatrix}\boldsymbol{p}_w^{i\;T} & \boldsymbol{v}_w^{i\;T} & \boldsymbol{q}_w^{i\;T} & \boldsymbol{b}_g^{\;T} & \boldsymbol{b}_a^{\;T}\end{bmatrix}^T \tag{2}$$

include the position, velocity and orientation of the IMU frame $\{i\}$ with respect to the world frame $\{w\}$, the gyroscope biases $\boldsymbol{b}_g$ and the accelerometer biases $\boldsymbol{b}_a$. We use rotation quaternions to model orientations.

The vision states

$$\boldsymbol{x}_V = \begin{matrix}\big[\boldsymbol{p}_w^{c_1\;T} & \dots & \boldsymbol{p}_w^{c_M\;T} & \boldsymbol{q}_w^{c_1\;T} & \dots & \boldsymbol{q}_w^{c_M\;T} \\ & & & \boldsymbol{f}_1^{\;T} & \dots & \boldsymbol{f}_N^{\;T}\big]^T\end{matrix} \tag{3}$$

include the orientations $\{\boldsymbol{q}_w^{c_i}\}_i$ and positions $\{\boldsymbol{p}_w^{c_i}\}_i$ of the camera frame at the last M image time instances, along with the 3D coordinates of N features $\left\{\boldsymbol{f}_j\right\}_j$. Each feature state $\boldsymbol{f}_j = \begin{bmatrix}\alpha_j & \beta_j & \rho_j\end{bmatrix}^T$ represents the inverse-depth parametrization of world feature $\boldsymbol{p}_j$ with respect to an anchor pose $\left\{c_{i_j}\right\}$ selected from the sliding window of poses. Inverse-depth is used due to its improved depth convergence properties for SLAM [31].

We model the dynamics of our system with the time-varying IMU measurements

$$\begin{cases} \dot{\boldsymbol{p}}_w^i &= \boldsymbol{v}_w^i \\ \dot{\boldsymbol{v}}_w^i &= \boldsymbol{C}(\boldsymbol{q}_w^i)^T(\boldsymbol{a}_{IMU} - \boldsymbol{b}_a - \boldsymbol{n}_a) + {}^w\boldsymbol{g} \\ \dot{\boldsymbol{q}}_w^i &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_{IMU} - \boldsymbol{b}_g - \boldsymbol{n}_g)\boldsymbol{q}_w^i \\ \dot{\boldsymbol{b}}_g &= \boldsymbol{n}_{b_g} \\ \dot{\boldsymbol{b}}_a &= \boldsymbol{n}_{b_a} \end{cases} \tag{4}$$

where $\boldsymbol{n}_\star$ are zero-mean Gaussian white noise disturbing the IMU measurements and modeling the biases as a random walk, and $\boldsymbol{C}(\boldsymbol{q})$ denotes the coordinate change matrix associated to a quaternion $\boldsymbol{q}$. The operator $\boldsymbol{\Omega}$ is defined by

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix}0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\lfloor\boldsymbol{\omega}\times\rfloor\end{bmatrix}, \tag{5}$$

$$\text{where } \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix}0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0\end{bmatrix}. \tag{6}$$

The state estimate is propagated at IMU rate and at first order using [32], [33].

We can express the cartesian coordinates of feature $\boldsymbol{p}_j$ in camera frame $\{c_i\}$ as

$$\begin{aligned} {}^{c_i}\boldsymbol{p}_j &= \begin{bmatrix}{}^{c_i}x_j & {}^{c_i}y_j & {}^{c_i}z_j\end{bmatrix}^T & (7) \\ &= \boldsymbol{C}(\boldsymbol{q}_w^{c_i})\left(\boldsymbol{p}_w^{c_{i_j}} + \frac{1}{\rho_j}\boldsymbol{C}(\boldsymbol{q}_w^{c_{i_j}})^T\begin{bmatrix}\alpha_j \\ \beta_j \\ 1\end{bmatrix} - \boldsymbol{p}_w^{c_i}\right). & (8) \end{aligned}$$

The associated visual measurement in SLAM is the pinhole projection of $\boldsymbol{p}_j$ over the normalized image plane $f = 1$ of camera frame $\{c_i\}$

$${}^i\boldsymbol{z}_j = \frac{1}{{}^{c_i}z_j}\begin{bmatrix}{}^{c_i}x_j \\ {}^{c_i}y_j\end{bmatrix} + {}^i\boldsymbol{n}_j \tag{9}$$

where ${}^i\boldsymbol{n}_j$ is a zero-mean white Gaussian feature measurement noise. This measurement can be linearized about the state estimate $\hat{\boldsymbol{x}}$

$$\begin{aligned} {}^i\delta\boldsymbol{z}_j \simeq \; & \boldsymbol{H}_{\boldsymbol{p}_{i_j}}\delta\boldsymbol{p}_w^{c_{i_j}} + \boldsymbol{H}_{\boldsymbol{p}_i}\delta\boldsymbol{p}_w^{c_i} + \boldsymbol{H}_{\boldsymbol{\theta}_{i_j}}\delta\boldsymbol{\theta}_w^{c_{i_j}} \\ & + \boldsymbol{H}_{\boldsymbol{\theta}_i}\delta\boldsymbol{\theta}_w^{c_i} + \boldsymbol{H}_{\boldsymbol{f}_j}\delta\boldsymbol{f}_j + {}^i\boldsymbol{n}_j \end{aligned} \tag{10}$$

with

$$\boldsymbol{H}_{\boldsymbol{p}_{i_j}} = {}^i\boldsymbol{J}_j\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_i}) \tag{11}$$

$$\boldsymbol{H}_{\boldsymbol{p}_i} = -{}^i\boldsymbol{J}_j\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_i}) \tag{12}$$

$$\boldsymbol{H}_{\boldsymbol{\theta}_{i_j}} = -\frac{1}{\hat{\rho}_j}{}^i\boldsymbol{J}_j\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_i})\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_{i_j}})^T\left\lfloor\begin{bmatrix}\hat{\alpha}_j \\ \hat{\beta}_j \\ 1\end{bmatrix}\times\right\rfloor \tag{13}$$

$$\boldsymbol{H}_{\boldsymbol{\theta}_i} = \begin{aligned} {}^i\boldsymbol{J}_j \bigg\lfloor \boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_i})\bigg(& \hat{\boldsymbol{p}}_w^{c_{i_j}} - \hat{\boldsymbol{p}}_w^{c_i} \\ & + \frac{1}{\hat{\rho}_j}\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_{i_j}})^T\begin{bmatrix}\hat{\alpha}_j \\ \hat{\beta}_j \\ 1\end{bmatrix}\bigg)\times\bigg\rfloor \end{aligned} \tag{14}$$

$$\boldsymbol{H}_{\boldsymbol{f}_j} = \frac{1}{\hat{\rho}_j}{}^i\boldsymbol{J}_j\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_i})\boldsymbol{C}(\hat{\boldsymbol{q}}_w^{c_{i_j}})^T\begin{bmatrix}1 & 0 & -\frac{\hat{\alpha}_j}{\hat{\rho}_j} \\ 0 & 1 & -\frac{\hat{\beta}_j}{\hat{\rho}_j} \\ 0 & 0 & -\frac{1}{\hat{\rho}_j}\end{bmatrix} \tag{15}$$

where ${}^i\boldsymbol{J}_j = \frac{1}{{}^{c_i}\hat{z}_j}\begin{bmatrix}\boldsymbol{I}_2 & -\hat{\boldsymbol{z}}_j\end{bmatrix}$. Note that when $i = i_j$, ${}^{i_j}\delta\boldsymbol{z}_j = \begin{bmatrix}\delta\alpha_j \\ \delta\beta_j\end{bmatrix}$. Since SLAM measurements are applied at every frame, they correspond to the latest pose in the sliding window and in practice $i = M$. As poses enter and exit the sliding window, ultimately the anchor pose $\left\{c_{i_j}\right\}$ will not be part of the state vector any more. To prevent this and keep using Eq. (10) in the EKF update, we reparametrize $\boldsymbol{p}_j$ to a new anchor at the start of the sliding window as in [34].

In our implementation, the pose states may also be used to perform MSCKF updates in addition to SLAM similarly to [35]. However, MSCKF with a *first-in first-out* sliding window scheme is only useful when the state poses cover a large enough baseline, which is not the case at standard image frame rate and UAV speeds [36]. It was therefore not used in our testing, and is not detailed here. Our complete implementation is fully detailed with Jacobians either referenced, or derived in [37].

IV. TEST FLIGHTS

A. Hardware

The quadrotor test platform in Figure 3 was built with parts available off the shelf. It is based on a Lumenier QAV400 airframe with the motors, propellers, and electronic speed controllers suggested by Lumenier. We selected Pixhawk Mini as our autopilot and set it up with the PX4 flight stack [38]. The embedded computer is a Nvidia Jetson TX2 module, which comes equipped with 6 ARMv8 CPU cores running at up to 2.5 GHz, and 8GB of memory.

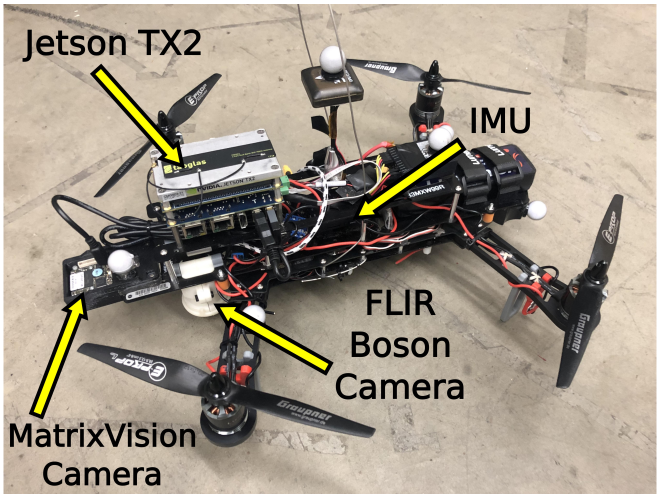


Fig. 3: Quadrotor platform equipped with a FLIR Boson thermal camera, a MATRIX VISION mvBlueFOX-MLC visible camera, a ICM20608 IMU inside a Pixhawk Mini autopilot, and a Nvidia Jetson TX2 computer.

Our thermal sensor is the industrial version of the FLIR Boson camera core with with a lens providing a 95-deg field of view (FOV). It delivers 640×512 images in the $7.5\mu m - 13.5\mu m$ longwave infrared spectral range, with a thermal sensitivity under 40 mK and a 8-ms time constant. The visible camera is a MATRIX VISION mvBlueFOX-MLC200WC equipped with a ~ 140-deg FOV lens. We used the ICM20608 IMU on board the Pixhawk mini.

The thermal camera was calibrated with the Kalibr toolbox [39], after leaving the target approximately 5 minutes in the sun. The target uses a circle grid pattern laminated over a Gatorfoam board.

B. Software settings

All our software runs on Ubuntu 16.04 with ROS Kinetic. Both camera drivers were set to grab 8-bit images at 30 fps with auto-exposure. The IMU runs at 200 Hz. The state estimation and autonomy software run in real time on the Jetson TX2.

Table I specifies the front end settings that were modified from the default values in OpenCV 3.3.1 for FAST detection and KLT tracking. The state estimator was initialized at zero-velocity, level attitude, and feature initial depth at the default commanded waypoint height with uncertainty matching the observed terrain distribution (± 15 cm, 3σ).

TABLE I: Front-end settings.

FAST detection threshold	20
Neighborhood margin (pixels)	40
KLT window size (pixels)	7x7
KLT max pyramidal level	3
KLT minimum eigen threshold	0.003
RANSAC error threshold (pixels)	1.0
RANSAC probability threshold	0.999

C. Autonomy Framework

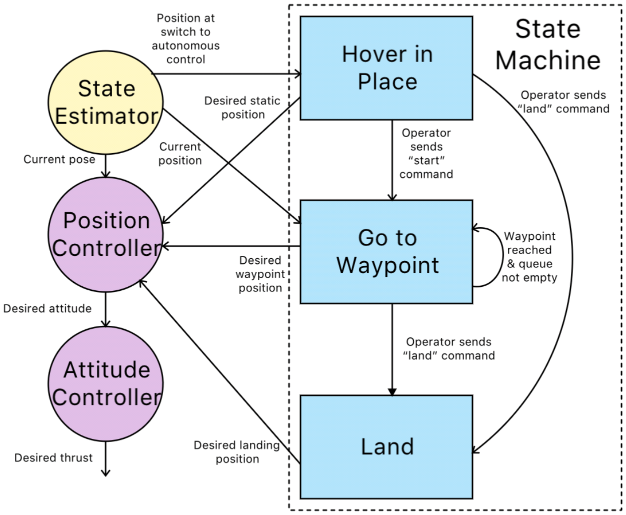


Fig. 4: Diagram of the state machine for closed-loop flights.

Figure 4 illustrates the state machine that was used to navigate the environment through a set of known waypoints and commit to a landing. The idle state upon startup commands the quadrotor to hover in place. Once the operator sends a *start* command, the state switches to *waypoint-following*, which specifies the trajectory for our test flights. It interpolates a trajectory between each waypoint and sends setpoints to the position controller, which runs independent of the state machine. When the last waypoint is reached, we send the vehicle on a vertical trajectory down to land.

We use the popular cascaded control architecture as used in [40]. The position controller runs on the TX2 and uses a PID-based feedback control policy with feedforward terms from the trajectory generated from the waypoint state. The position controller sends the desired attitude and thrust commands to the PX4 flight stack which runs a globally stable non-linear attitude controller based on unit quaternions [41] on the Pixhawk Mini. It generates a desired thrust per motor and sends commands to the ESCs to control the angular velocities of the motors.

D. Closed-loop controlled flights

We demonstrated outdoor autonomous flight with waypoint navigation and landing at night in the Mars Yard at JPL using the framework in Figure 4. Position and velocity estimates from thermal-inertial odometry were used to close the control loop.

The state estimator was initialized with the quadrotor on a platform 50cm tall, to allow for the thermal camera to see some features on the ground upon startup. A safety pilot would take off manually and maintain hover flight, and switch to autonomous controls if state estimator and control outputs were nominal.

The quadrotor would then proceed to follow a 2m×2m square pattern and attempt to land on a 1m×1m landing pad at the end of it. While ground truth was not available at the time of these tests, the system demonstrated 5 successful flights, and landing on the platform. The video in the supplemental materials of this paper shows one of these flights.

V. PERFORMANCE ANALYSIS

A. *Dataset with ground truth*

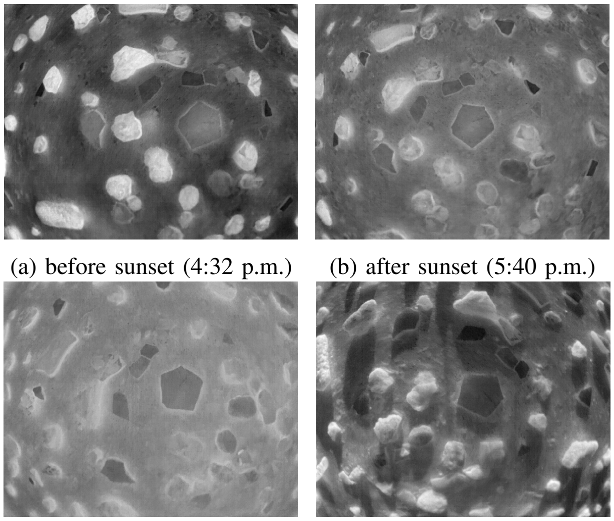
We collected an extensive dataset to characterize the sensitivity of the thermal-inertial state estimation demonstrated in Subsection IV-D to the variation of terrain appearance in the thermal waveband for various times of the night. We also collected an independent dataset to evaluate the effect of the time elapsed since the last FFC at the same time of the night.

The ground truth pose was provided by 7 OptiTrack Prime41 motion capture cameras at 120 Hz. One can generally expect millimeter position and sub-degree orientation performance. Even though these cameras are branded for outdoor capture in sunlight, the pose output showed outliers around sunset and sunrise when the sun was in the field of view. To eliminate these, we filtered the OptiTrack pose with the IMU. Since our state estimation is using the same IMU, one could argue that our ground truth is no longer independent. However, since ground truth accuracy is much better than IMU-only accuracy, sharing the same IMU between the navigation and ground truth filters still allows reasonable comparison of errors. These flights used the autonomy framework from Section IV-C with the quadrotor flown with closed-loop controls. We used the ground truth motion capture system as a position control input to guarantee that the trajectory flown and the scene observed remain the same independent of thermal-inertial odometry errors.

The first dataset includes flights approximately every hour from 1 hour before sunset until 2 hours after sunrise. A FFC was performed on the camera before each flight so as to keep its effect separate. Figure 5 illustrates the various appearances of the same terrain throughout the night on a thermal camera.

The second dataset includes flights every 5 minutes without a FFC being triggered, and up to 34 mins after FFC. We attempted to collect this dataset as much as possible at the same time of the night. However since the battery had to be swapped, we had to wait to resume at the same time since FFC, and the overall dataset spans 52 minutes.

This dataset was acquired between February 7-8, 2019 in Pasadena, CA. The sun set and rose at 5:29 p.m. and 6:44 a.m., respectively. The weather was sunny on both days. For all the results in this section, state estimation is initialized at the beginning of the 2m square pattern, without prior excitation.



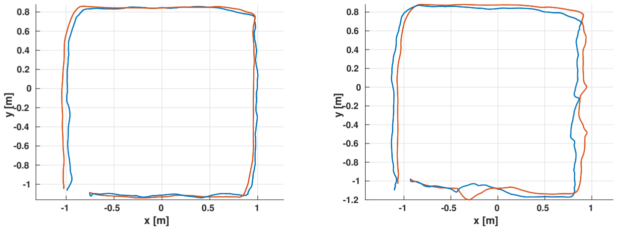
(a) before sunset (4:32 p.m.)   (b) after sunset (5:40 p.m.)

(c) before sunrise (6:30 a.m.)   (d) after sunrise (7:53 a.m.)

Fig. 5: Variation of thermal image appearance throughout the night. Sunset and sunrise provide the most sudden changes.

B. *TIO vs. VIO*

Visual-inertial odometry performance is a useful reference to evaluate thermal-inertial odometry performance on our system. In addition to the two datasets, we performed a reference flight with the visible camera during daytime. The position estimate of our VIO state estimator is compared to TIO running with the same parameters in Figure 6.



(a) TIO at night time   (b) VIO at daytime

Fig. 6: Top view of the trajectory estimate (blue) and ground truth (red) at night (6:30 a.m.) with thermal camera (left), and during the day (8:41 a.m.) with the visible camera (right).

For this comparison, we picked the thermal sequence in which the images seemed to have the least amount of contrast to stress our system. It was acquired at 6:30 a.m., just before the local sunrise, after the ground had cooled off all night. At the end of the sequence, TIO had drifted 1.6% of the distance, versus 2.1% for VIO. While state-of-the-art VIO results can be under 1%, we believe this performance is on par given the absence of excitation, the IMU noise characteristics, and the vibration levels during flight. A comparison with the rest of literature is the subject of on-going work as it requires dedicated understanding and adaptation of each algorithm to be fair. Overall, although thermal infrared imagery provides typically low contrast images with less features, in our results it provided odometry constraints at nighttime as efficient as vision at daytime.

C. State Estimation versus Time of Day

Figure 7 shows the final position error drift over the same trajectory throughout the night.

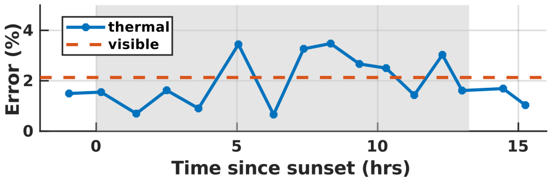


Fig. 7: Thermal-inertial final position estimation error drift at various times of the night. The period between local sunset and sunrise is shown in gray. VIO performance is shown for reference.

Drift remains within 3.5% on all flights. We were able to correlate all errors above 2% to trajectories where the height controlled through OptiTrack was up to 70 cm below the 2m reference height, due to to the loose gains the controller was unfortunately set to. When flying closer to the ground, the authors noticed the feature depth uncertainty had a larger impact.

With proper settings, there seems to be no correlation between the time of the night and the thermal-inertial odometry performance. Although the image contrast goes down, {FAST+KLT} can still be set to track correctly. We highlight the importance of raising the FAST feature threshold, as well as the minimum eigenvalue in KLT with respect to VIO. In our setting, the minimum eigenvalue had to be tripled with respect to the default value used in VIO. Without this change, a significant number of FAST features would drift erroneously with KLT in the thermal image.

D. State Estimation versus Time since FFC

Figure 8 shows the evolution of the TIO performance up to 34 min since the last FFC. The dataset was acquired between 10:32 p.m. and 11:24 p.m.

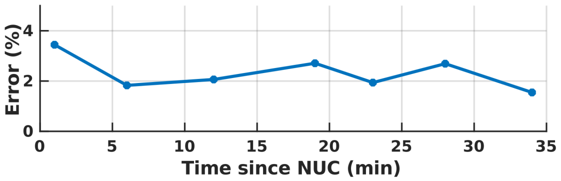
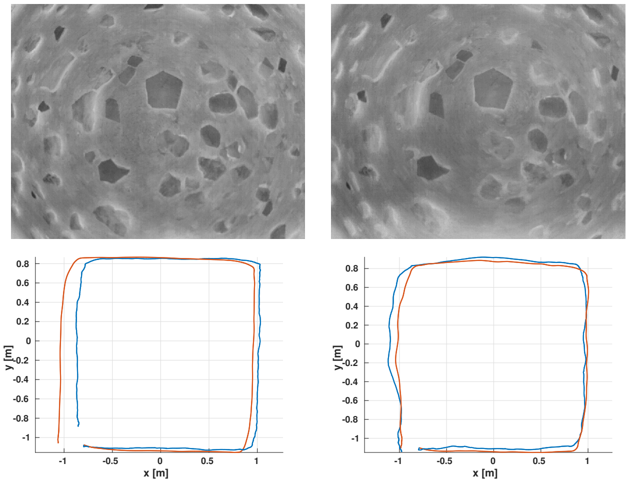


Fig. 8: Thermal-inertial position estimation error for various times since FFC.

We note no evident correlation between performance and FFC. The performance at FFC + 1 min is slightly worse, but that is also the closest flight to the ground of the two datasets and has the worst overall performance. For the same reason mentioned in Subsection V-C, we believe this performance variation is unrelated to the FFC itself. Figure 9 shows a sample image and the trajectory reconstruction immediately after FFC, and 34 min later.



(a) NUC + 1 min

(b) NUC + 34 min

Fig. 9: Sample thermal image after FFC + 1 min, and at FFC + 34 min (top). Corresponding position estimate in blue and ground truth in red (bottom). In our sequences, not proceeding to a FFC led to a white veil on the bottom side of the image, which does not affect state estimation over the battery flight time.

At FFC + 34 min, one can notice the image contrast drops, especially at the bottom edge of the field of view. However, this did not represent a large view angle and feature tracking appears unaffected in our tests. Our TIO solution can survive without features for the duration of a FFC on the Boson camera (∼0.25s), but this is an important result for thermal-only odometry solutions that typically break during FFC in the absence of loop closure, as all feature tracks are lost. During the lifetime of our quadrotor battery, which is less than 30 min, thermal-based navigation is not affected by non-uniformities.

VI. Conclusion

This paper has presented a state estimation architecture that can perform TIO with a thermal-spectrum camera, or VIO with a visible-spectrum camera. We highlighted the importance of raising the FAST feature detection threshold, as well as the minimum eigenvalue in KLT with respect to VIO. Our state estimation was integrated inside an autonomy framework which runs in real time on-board an off-the-shelf drone. We demonstrated closed-loop controlled flight with waypoint navigation and landing at nighttime with the thermal camera. An extensive dataset was collected for various times of the night, and various times since FFC. TIO performed similarly throughout the night, and on par with VIO during the day. TIO was not affected by the thermal sensor non-uniformities during at least the 34 min that we tested, which is longer than most drone batteries last.

These results indicate that TIO is a viable alternative to GPS for autonomous flights at night time, just like VIO at daytime. However, TIO is just in its infancy and many areas need to be characterized or improved. Extensive testing similar to the one presented here needs to be conducted over various types of terrain, and various weather conditions. Thermal feature tracks with our {FAST+KLT} front end appeared at least less numerous than visual tracks over the same scene. This did not affect performance in our flights, which happened at relatively low speed ($\sim$ 0.3 m/s). However, demonstrating aggressive maneuvers at night may require different processing, e.g. due to the rolling shutter read-out architecture or relatively long time constant ($\sim$ 8 ms) of uncooled thermal cameras.

ACKNOWLEDGMENT

The authors would like to thank Sawyer Brooks, Shreetej Reddy, Elvin Gu, Daniel Nugent, Tobias Shapinsky and Amanda Bouman for their help with hardware, software, and dataset collection.

REFERENCES

[1] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[2] R. Gade and T. B. Moeslund. Thermal cameras and applications: A survey. *Mach. Vision Appl.*, 25(1):245–262, 2014.

[3] J.T. Ward. *Realistic Texture in Simulated Thermal Infrared Imagery*. PhD thesis, Rochester Institute of Technology, 2008.

[4] FLIR. Boson thermal imaging core engineering datasheet. Technical report, FLIR, 2018.

[5] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, 2007.

[6] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 957–964, 2012.

[7] J. A. Delmerico and D. Scaramuzza. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2502–2509, 2018.

[8] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3565–3572, 2007.

[9] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[10] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[11] L. von Stumberg, V. C. Usenko, and D. Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2510–2517, 2018.

[12] T. B. Schön and J. Roll. Ego-motion and indirect road geometry estimation using night vision. In *IEEE Intelligent Vehicles Symposium, Proceedings*, 2009.

[13] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[14] P. V. K. Borges and S. Vidas. Practical infrared visual odometry. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2205–2213, 2016.

[15] C. Brunner, T. Peynot, T. Vidal-Calleja, and J. Underwood. Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *Journal of Field Robotics*, 30(4):641–666, 2013.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[17] L. Chen, L. Sun, T. Yang, L. Fan, K. Huang, and Z. XuanYuan. Rgb-t slam: A flexible slam framework by combining appearance and thermal information. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5682–5687, 2017.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. pages 2564–2571, 11 2011.

[19] S. Vidas and S. Sridharan. Hand-held monocular slam in thermal-infrared. In *12th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2012.

[20] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2010.

[21] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 594–600, 1994.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[23] T. Mouats, N. Aouf, L. Chermak, and M. A. Richardson. Thermal stereo odometry for uavs. *IEEE Sensors Journal*, 15(11):6335–6347, 2015.

[24] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012.

[25] S. Khattak, C. Papachristos, and K. Alexis. Keyframe-based direct thermal-inertial odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[26] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[27] J. P. Lewis. Fast template matching. In *Vision interface*, volume 95, pages 15–19, 1995.

[28] S. K. Kim, D. Gallup, J. Frahm, and M. Pollefeys. Joint radiometric calibration and feature tracking for an adaptive stereo system. *Computer Vision Image Understanding*, 114:574–582, May 2010.

[29] P. Bergmann, R. Wang, and D. Cremers. Online photometric calibration of auto exposure video for realtime visual odometry and slam. *IEEE Robotics and Automation Letters*, 3(2):627–634, 2018.

[30] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[31] J. Civera, A. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.

[32] S. Weiss and R. Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *IEEE International Conference on Robotics and Automation*, 2012.

[33] N. Trawny and S.I. Roumeliotis. Indirect Kalman Filter for 3D Attitude Estimation. Technical report, University of Minnesota, 2005.

[34] M. Li and A.I. Mourikis. Optimization-Based Estimator Design for Vision-Aided Inertial Navigation: Supplemental Materials. In *Robotics: Science and Systems*, 2012.

[35] M. Li and A.I. Mourikis. Optimization-Based Estimator Design for Vision-Aided Inertial Navigation. In *Robotics: Science and Systems*, 2012.

[36] Dimitrios G. Kottas, Kejian Wu, and Stergios I. Roumeliotis. Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3172–3179, 2013.

[37] J. Delaune and D.S. Bayard. xvio: Visual-inertial odometry implementation. Technical report, Jet Propulsion Laboratory (available upon request), 2019.

[38] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *IEEE International Conference on Robotics and Automation*, pages 2992–2997, 2011.

[39] Paul Timothy Furgale, Jörn Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286, 2013.

[40] T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *49th IEEE conference on decision and control (CDC)*, pages 5420–5425, 2010.

[41] D. Brescianini, M. Hehn, and R. D'Andrea. Nonlinear quadrocopter attitude control: Technical report. Technical report, ETH Zurich, 2013.